Task_2_Prodigy_Internship

June 4, 2024

PRODIGY INFOTECH DATA SCIENCE INTERN

#TASK 2

TASK OVERVIEW: Perform data cleaning and exploratory data analysis (EDA) on a dataset of your choice, such as the Titanic dataset from Kaggle. Explore the relationships between variables and identify patterns and trends in the data.

```
[ ]: #Here import the necessary libraries for this task

     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
```

Importing the Titanic dataset here.

```
[ ]: df = pd.read_csv("/content/titanic dataset.csv")
```

Data Prepocessing and Data Cleaning

```
[ ]: df.head()
```

```
[ ]:    PassengerId  Survived  Pclass  \
     0          892         0       3
     1          893         1       3
     2          894         0       2
     3          895         0       3
     4          896         1       3

                                                Name     Sex   Age  SibSp  Parch  \
     0                              Kelly, Mr. James    male  34.5      0      0
     1              Wilkes, Mrs. James (Ellen Needs)  female  47.0      1      0
     2                     Myles, Mr. Thomas Francis    male  62.0      0      0
     3                              Wirz, Mr. Albert    male  27.0      0      0
     4  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1      1

         Ticket     Fare Cabin Embarked
     0   330911   7.8292   NaN        Q
     1   363272   7.0000   NaN        S
     2   240276   9.6875   NaN        Q
```

```
3   315154   8.6625   NaN        S
4  3101298  12.2875   NaN        S
```

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    int64
 2   Pclass       418 non-null    int64
 3   Name         418 non-null    object
 4   Sex          418 non-null    object
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64
 7   Parch        418 non-null    int64
 8   Ticket       418 non-null    object
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object
 11  Embarked     418 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

```
[ ]: df.isnull().sum()
```

```
[ ]: PassengerId      0
     Survived         0
     Pclass           0
     Name             0
     Sex              0
     Age             86
     SibSp            0
     Parch            0
     Ticket           0
     Fare             1
     Cabin          327
     Embarked         0
     dtype: int64
```

```
[ ]: df.columns
```

```
[ ]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
            'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
           dtype='object')
```

Remove the irrelevant columns

[ ]:

```
df1 = df.drop('Cabin', axis=1)  # Specify axis=1 for columns
df1
```

[ ]:

```
     PassengerId  Survived  Pclass  \
0            892         0       3
1            893         1       3
2            894         0       2
3            895         0       3
4            896         1       3
..           …       …     …
413         1305         0       3
414         1306         1       1
415         1307         0       3
416         1308         0       3
417         1309         0       3

                                             Name     Sex   Age  SibSp  Parch  \
0                                Kelly, Mr. James    male  34.5      0      0
1                Wilkes, Mrs. James (Ellen Needs)  female  47.0      1      0
2                       Myles, Mr. Thomas Francis    male  62.0      0      0
3                                Wirz, Mr. Albert    male  27.0      0      0
4    Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1      1
..                                            …     …   …    …    …
413                            Spector, Mr. Woolf    male   NaN      0      0
414                  Oliva y Ocana, Dona. Fermina  female  39.0      0      0
415                  Saether, Mr. Simon Sivertsen    male  38.5      0      0
416                           Ware, Mr. Frederick    male   NaN      0      0
417                      Peter, Master. Michael J    male   NaN      1      1

                 Ticket      Fare Embarked
0                330911    7.8292        Q
1                363272    7.0000        S
2                240276    9.6875        Q
3                315154    8.6625        S
4               3101298   12.2875        S
..                  …       …      …
413           A.5. 3236    8.0500        S
414            PC 17758  108.9000        C
415  SOTON/O.Q. 3101262    7.2500        S
416              359309    8.0500        S
417                2668   22.3583        C

[418 rows x 11 columns]
```

[ ]:

```
df.nunique()
```

```
[ ]: PassengerId    418
     Survived         2
     Pclass           3
     Name           418
     Sex              2
     Age             79
     SibSp            7
     Parch            8
     Ticket         363
     Fare           169
     Cabin           76
     Embarked         3
     dtype: int64
```

```
[ ]: df.duplicated()
```

```
[ ]: 0      False
     1      False
     2      False
     3      False
     4      False
            …
     413    False
     414    False
     415    False
     416    False
     417    False
     Length: 418, dtype: bool
```

```
[ ]: df.describe(include=['number'])
```

```
[ ]:        PassengerId    Survived      Pclass         Age       SibSp  \
     count   418.000000  418.000000  418.000000  332.000000  418.000000
     mean   1100.500000    0.363636    2.265550   30.272590    0.447368
     std     120.810458    0.481622    0.841838   14.181209    0.896760
     min     892.000000    0.000000    1.000000    0.170000    0.000000
     25%     996.250000    0.000000    1.000000   21.000000    0.000000
     50%    1100.500000    0.000000    3.000000   27.000000    0.000000
     75%    1204.750000    1.000000    3.000000   39.000000    1.000000
     max    1309.000000    1.000000    3.000000   76.000000    8.000000

                 Parch        Fare
     count  418.000000  417.000000
     mean     0.392344   35.627188
     std      0.981429   55.907576
     min      0.000000    0.000000
     25%      0.000000    7.895800
```

```
50%      0.000000   14.454200
75%      0.000000   31.500000
max      9.000000  512.329200
```

Handling the Missing Values

```
[ ]: df1['Fare'] = df['Fare'].fillna(df['Fare'].mean())
```

```
[ ]: df1['Age'] = df1['Age'].fillna(df['Age'].mean())
```

```
[ ]: df1.isnull().sum()
```

```
[ ]: PassengerId    0
     Survived       0
     Pclass         0
     Name           0
     Sex            0
     Age            0
     SibSp          0
     Parch          0
     Ticket         0
     Fare           0
     Embarked       0
     dtype: int64
```

EDA

```
[ ]: df1['PassengerId'].value_counts()
```

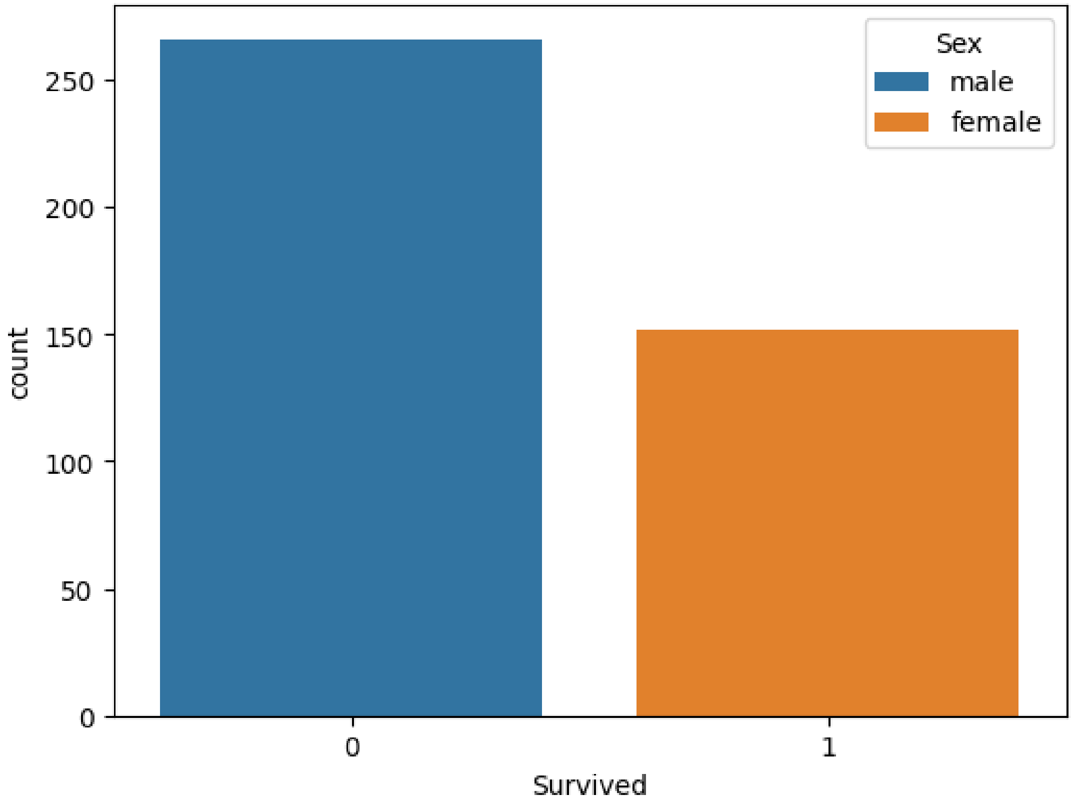
```
[ ]: PassengerId
     892     1
     1205    1
     1177    1
     1176    1
     1175    1
            ..
     1028    1
     1027    1
     1026    1
     1025    1
     1309    1
     Name: count, Length: 418, dtype: int64
```

```
[ ]: df1['Survived'].value_counts()
```

```
[ ]: Survived
     0    266
     1    152
```

```
Name: count, dtype: int64
```

```
[ ]: sns.countplot(x ='Survived' , hue = 'Sex' , data = df1)
     plt.show()
```

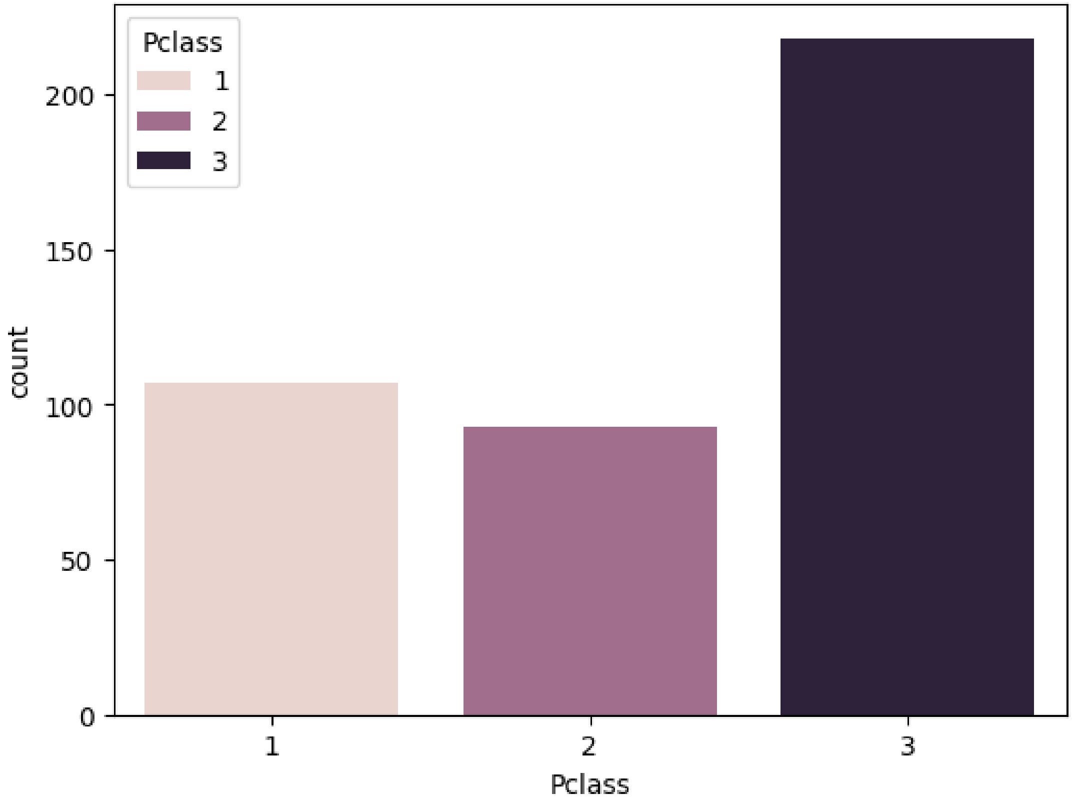


```
[ ]: df1['Pclass'].value_counts()
```

```
[ ]: Pclass
     3    218
     1    107
     2     93
     Name: count, dtype: int64
```

```
[ ]: sns.countplot(x='Pclass',hue='Pclass',data=df1)
```

```
[ ]: <Axes: xlabel='Pclass', ylabel='count'>
```

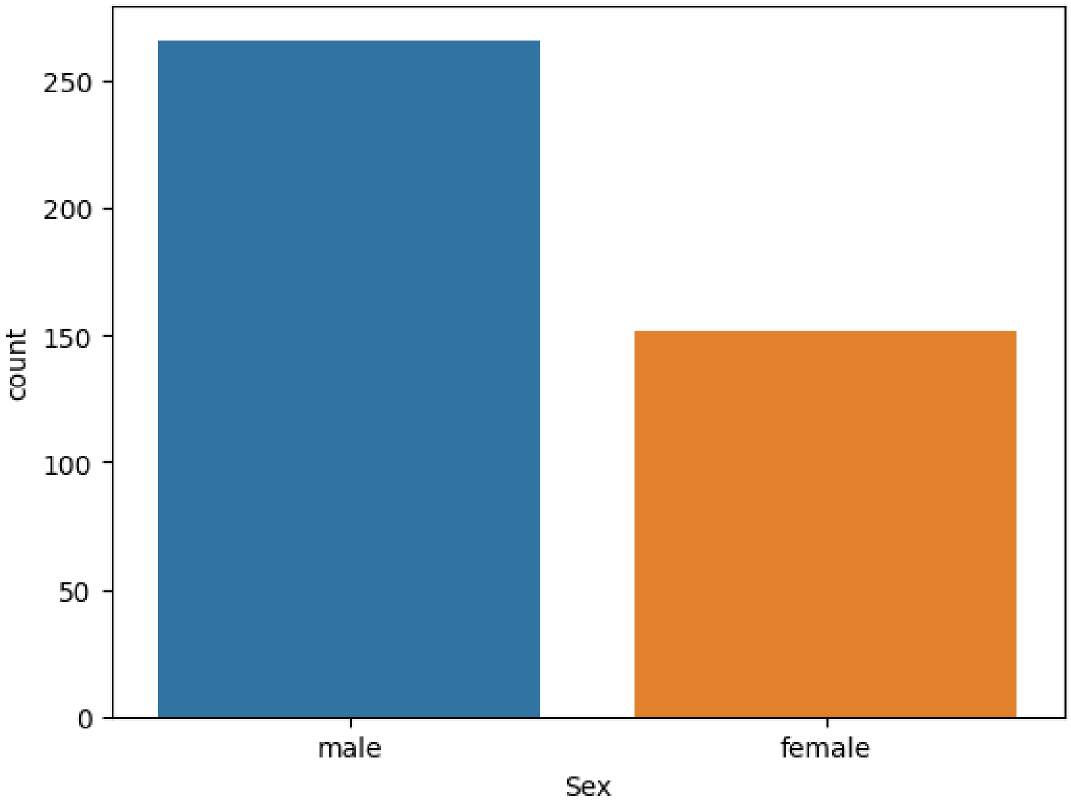


```
[ ]: gender_count = df1['Sex'].value_counts()
     gender_count
```

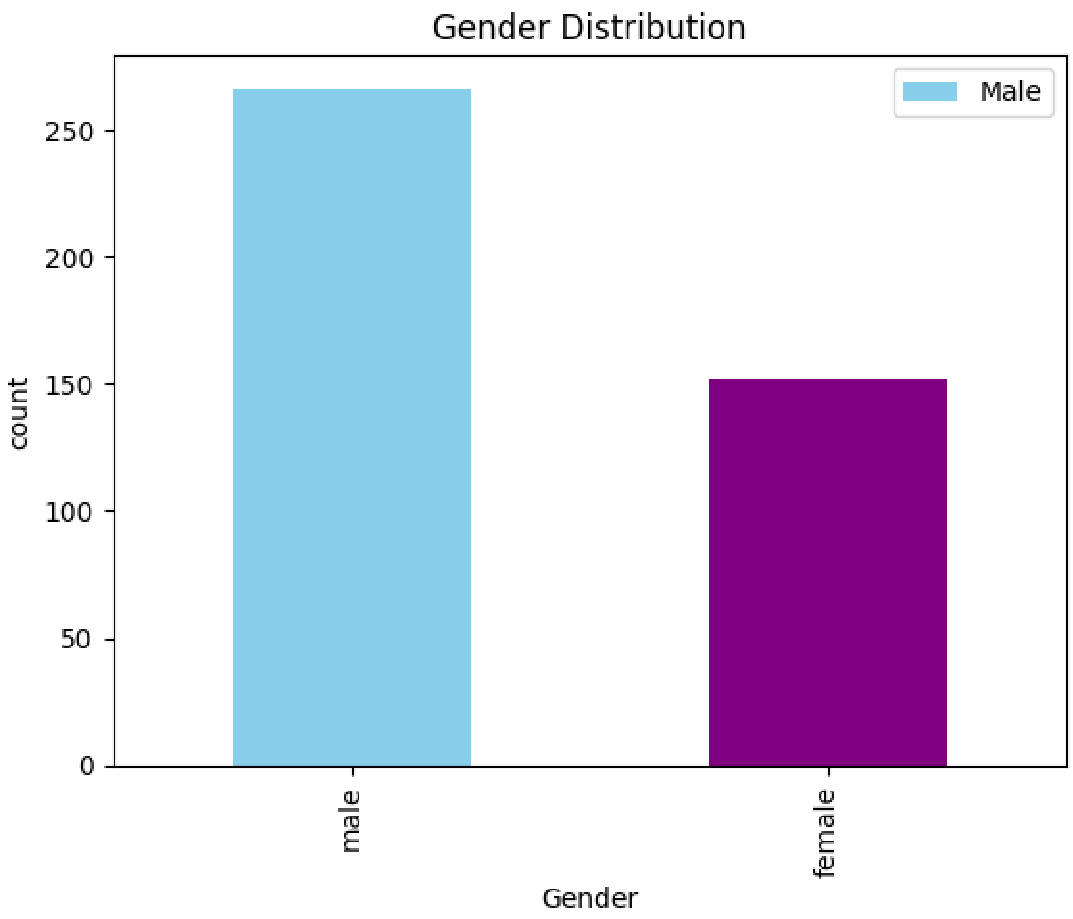
```
[ ]: Sex
     male      266
     female    152
     Name: count, dtype: int64
```

```
[ ]: sns.countplot (x='Sex',hue='Sex',data=df1)
```

```
[ ]: <Axes: xlabel='Sex', ylabel='count'>
```



[ ]:
```
plt.figure()
gender_count.plot(kind="bar",color=["skyblue","purple"])
plt.title("Gender Distribution")
plt.xlabel("Gender")
plt.ylabel("count")
plt.show()
```

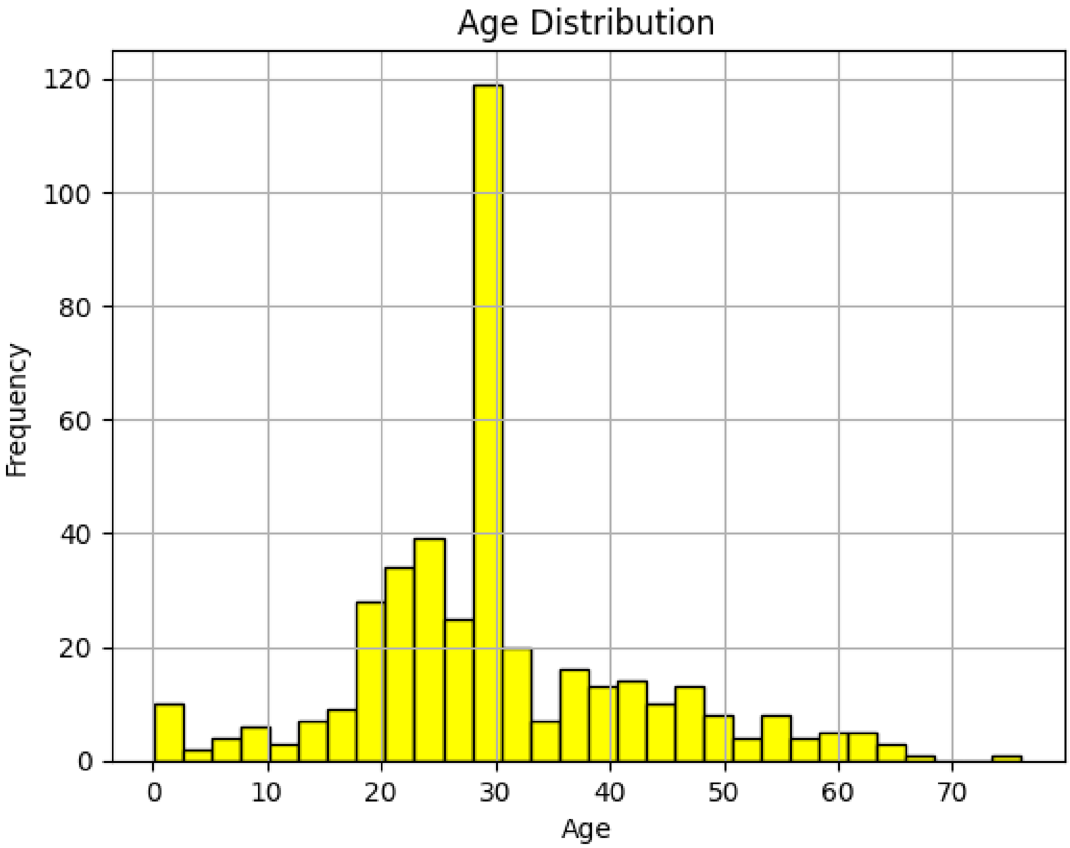


```
[ ]: df1.columns
```

```
[ ]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
              'Parch', 'Ticket', 'Fare', 'Embarked'],
             dtype='object')
```

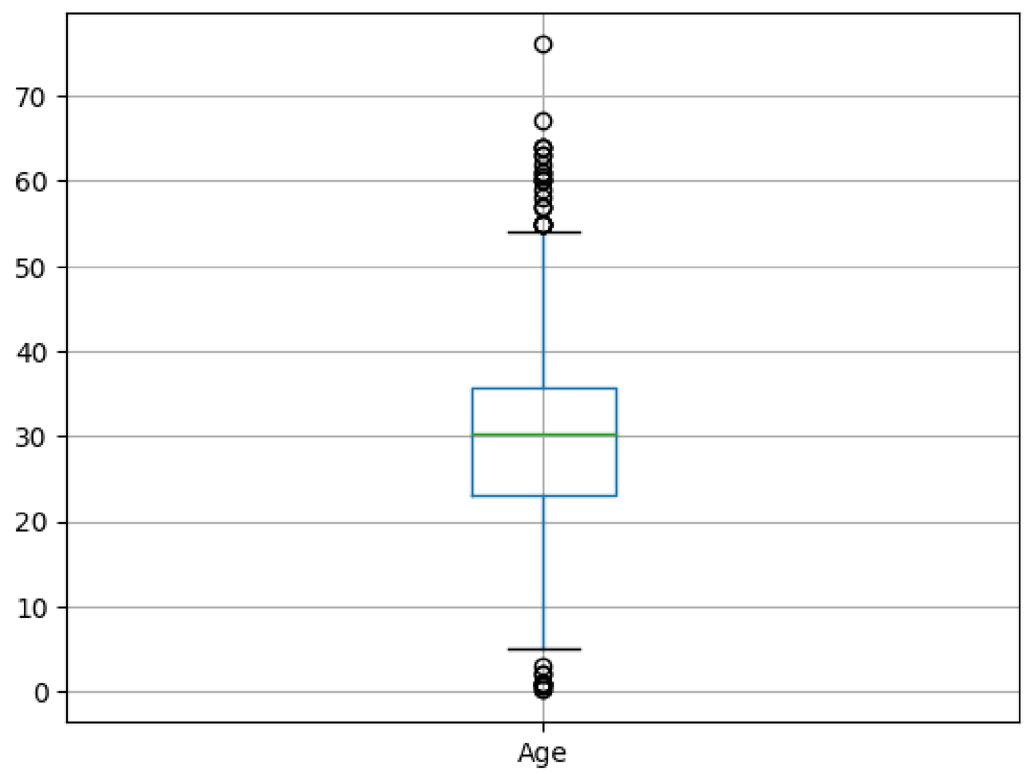
```
[ ]: df1['Age'].hist(bins=30,color="yellow",edgecolor="black")
     plt.title("Age Distribution")
     plt.xlabel("Age")
     plt.ylabel("Frequency")
```

```
[ ]: Text(0, 0.5, 'Frequency')
```



```
[ ]: df1[['Age']].boxplot()
```

```
[ ]: <Axes: >
```

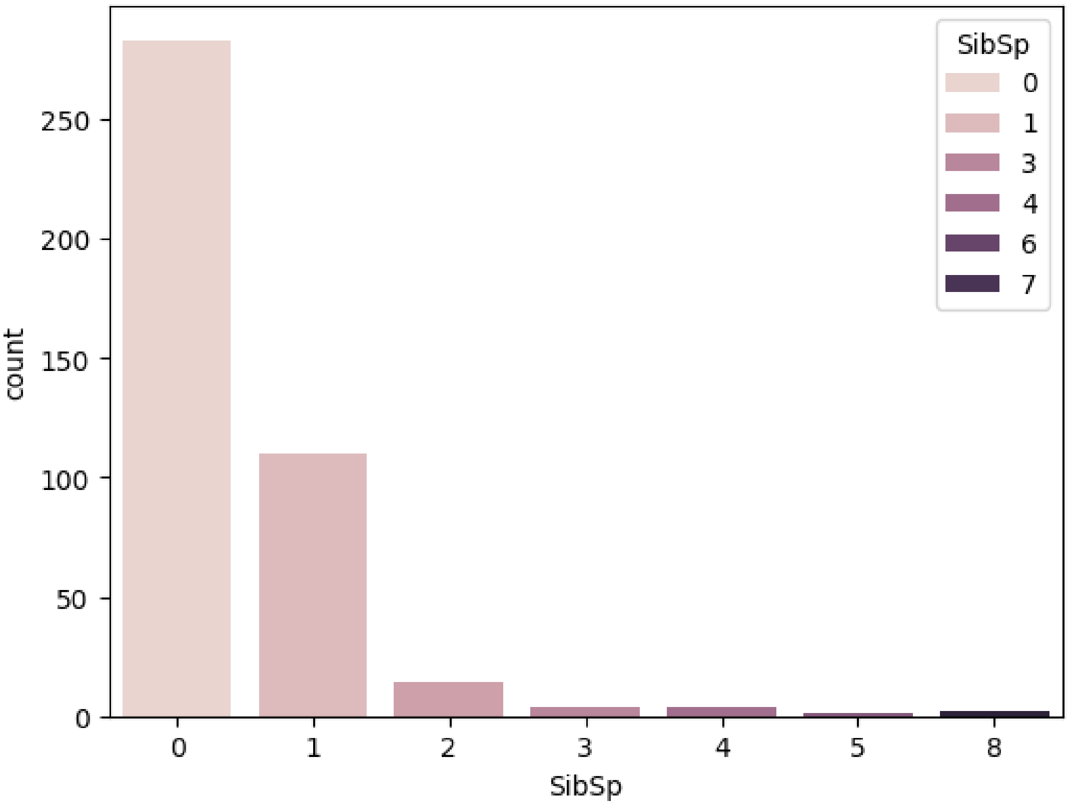


[ ]:
```
df1['SibSp'].value_counts()
```

[ ]:
```
SibSp
0    283
1    110
2     14
3      4
4      4
8      2
5      1
Name: count, dtype: int64
```

[ ]:
```
sns.countplot(x="SibSp",hue='SibSp',data=df1)
```

[ ]:
```
<Axes: xlabel='SibSp', ylabel='count'>
```

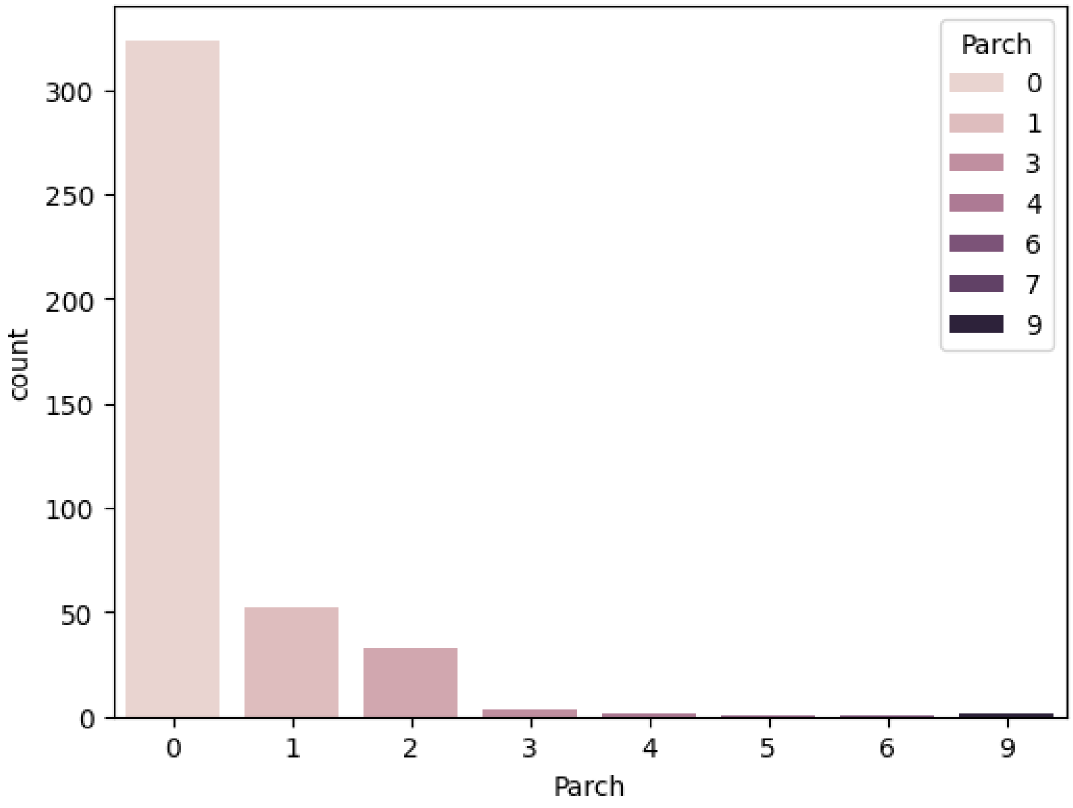


```
[ ]: df1['Parch'].value_counts()
```

```
[ ]: Parch
     0    324
     1     52
     2     33
     3      3
     4      2
     9      2
     6      1
     5      1
     Name: count, dtype: int64
```

```
[ ]: sns.countplot(x="Parch",hue='Parch',data=df1)
```

```
[ ]: <Axes: xlabel='Parch', ylabel='count'>
```

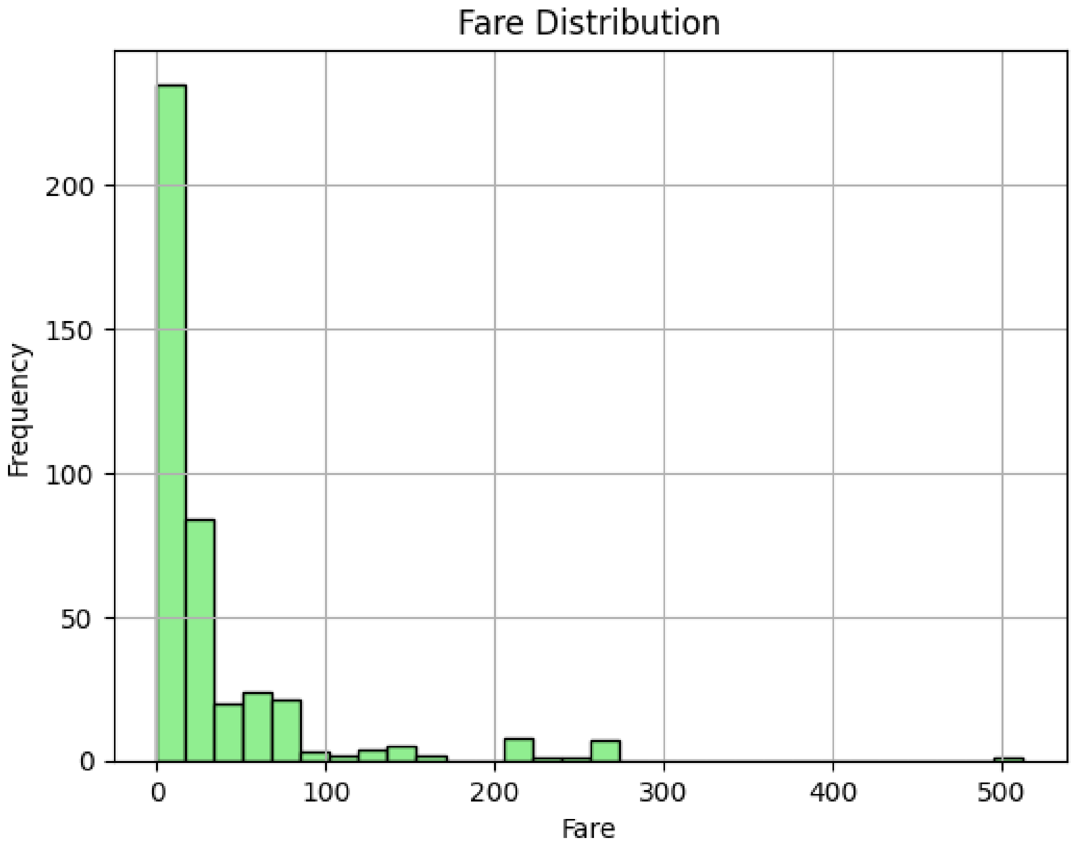


[ ]:

```
df1['Fare'].hist(bins=30,color="lightgreen",edgecolor="black")
plt.title("Fare Distribution")
plt.xlabel("Fare")
plt.ylabel("Frequency")
```

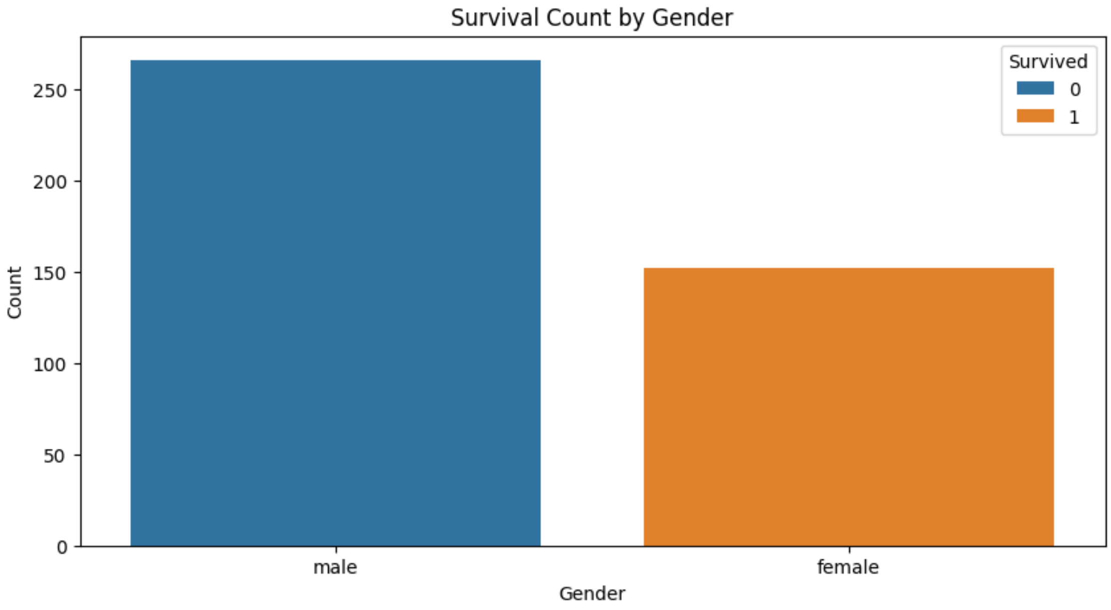
[ ]:

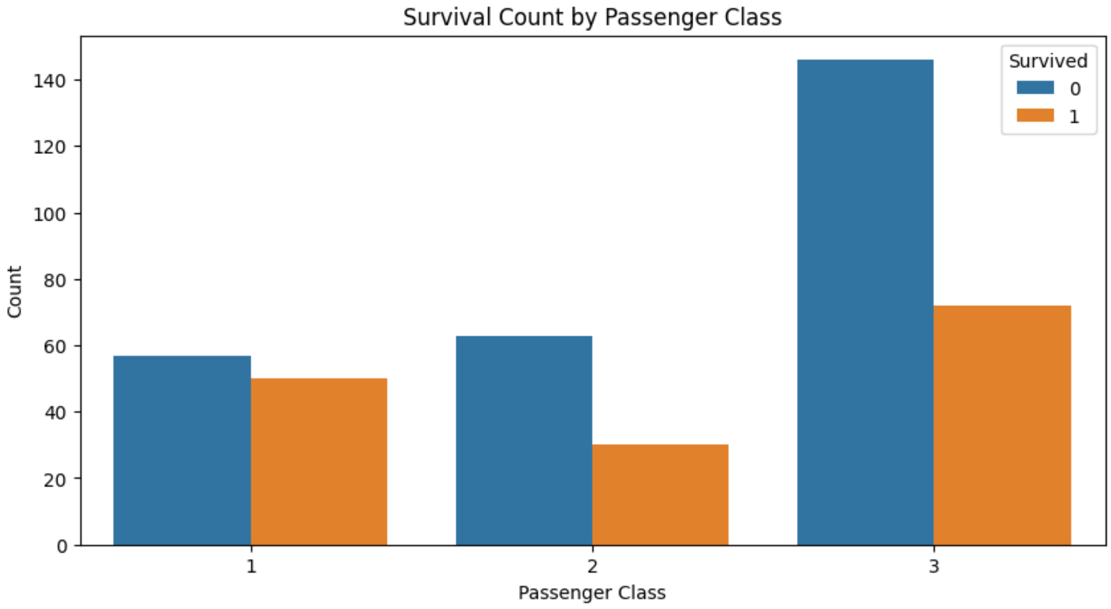
```
Text(0, 0.5, 'Frequency')
```



[ ]:

```
# Visualization
plt.figure(figsize=(10, 5))
sns.countplot(data=df1, x='Sex', hue='Survived')
plt.title('Survival Count by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()

plt.figure(figsize=(10, 5))
sns.countplot(data=df1, x='Pclass', hue='Survived')
plt.title('Survival Count by Passenger Class')
plt.xlabel('Passenger Class')
plt.ylabel('Count')
plt.show()
```





Correlation

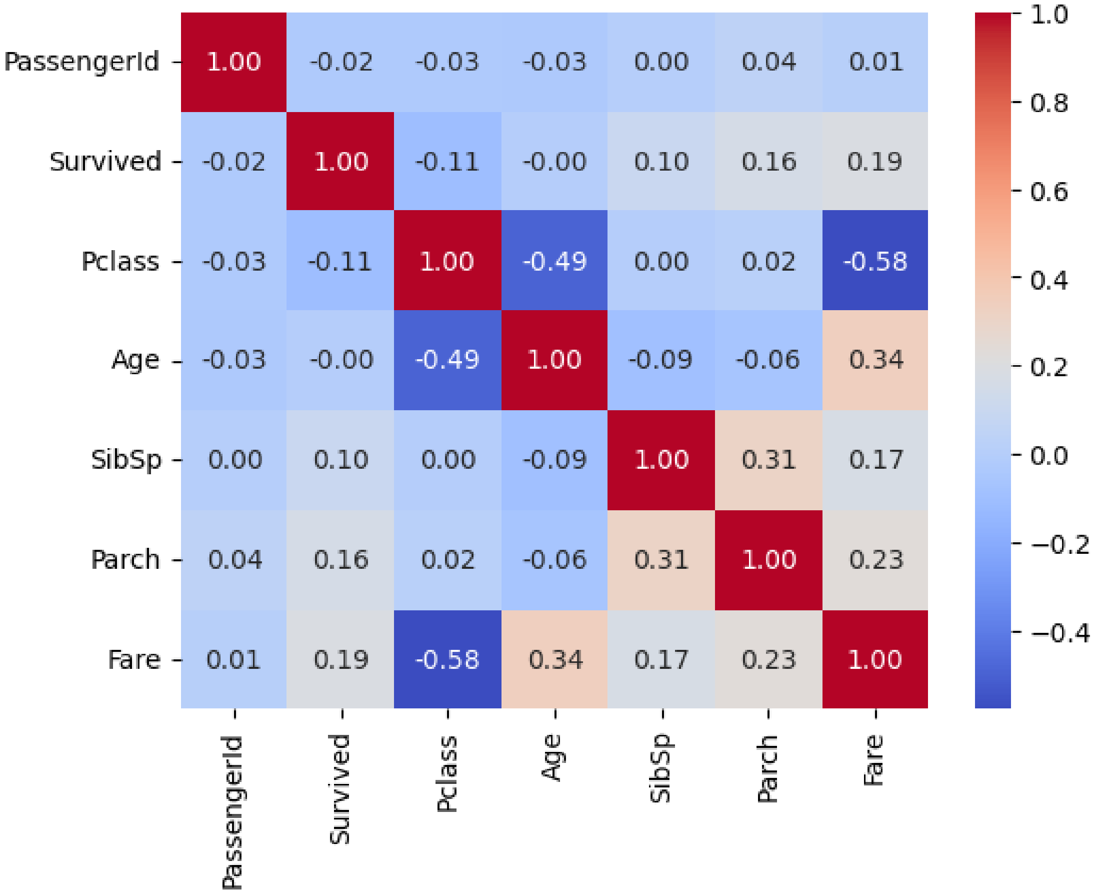
```
[ ]: df_number = df.select_dtypes(include=np.number)
```

```
[ ]: df_number
[ ]:      PassengerId  Survived  Pclass   Age  SibSp  Parch      Fare
     0            892         0       3  34.5      0      0    7.8292
     1            893         1       3  47.0      1      0    7.0000
     2            894         0       2  62.0      0      0    9.6875
     3            895         0       3  27.0      0      0    8.6625
     4            896         1       3  22.0      1      1   12.2875
     ..           …       …     …   …     …    …       …
     413         1305         0       3   NaN      0      0    8.0500
     414         1306         1       1  39.0      0      0  108.9000
     415         1307         0       3  38.5      0      0    7.2500
     416         1308         0       3   NaN      0      0    8.0500
     417         1309         0       3   NaN      1      1   22.3583

     [418 rows x 7 columns]
```

```
[ ]: df_number.corr()
```

```
[ ]:              PassengerId  Survived    Pclass       Age     SibSp     Parch  \
     PassengerId     1.000000 -0.023245 -0.026751 -0.034102  0.003818  0.043080
     Survived       -0.023245  1.000000 -0.108615 -0.000013  0.099943  0.159120
     Pclass         -0.026751 -0.108615  1.000000 -0.492143  0.001087  0.018721
     Age            -0.034102 -0.000013 -0.492143  1.000000 -0.091587 -0.061249
     SibSp           0.003818  0.099943  0.001087 -0.091587  1.000000  0.306895
     Parch           0.043080  0.159120  0.018721 -0.061249  0.306895  1.000000
     Fare            0.008211  0.191514 -0.577147  0.337932  0.171539  0.230046

                      Fare
     PassengerId  0.008211
     Survived     0.191514
     Pclass      -0.577147
     Age          0.337932
     SibSp        0.171539
     Parch        0.230046
     Fare         1.000000
```

```
[ ]: plt.figure()
     sns.heatmap(df_number.corr(),annot=True,cmap='coolwarm',fmt=".2f")
     plt.show()
```



Statistical Analysis

[ ]:

```
# Statistical Analysis
from scipy.stats import chi2_contingency

# Chi-square test for gender and survival
chi2_gender_survival = chi2_contingency(pd.crosstab(df1['Sex'],␣
  ↪df1['Survived']))
print("Chi-square Test for Gender and Survival:")
print(f"Chi-square value: {chi2_gender_survival[0]}")
print(f"P-value: {chi2_gender_survival[1]}")
```

```
Chi-square Test for Gender and Survival:
Chi-square value: 413.6897405343716
P-value: 5.767311139789629e-92
```

For the chi-square test between gender and survival: the p-value is greater than 0.05, it suggests that there is no significant relationship between gender and survival.

[ ]:

```
# Chi-square test for passenger class and survival
chi2_class_survival = chi2_contingency(pd.crosstab(df1['Pclass'],␣
 ↪df1['Survived']))
print("Chi-square Test for Passenger Class and Survival:")
print(f"Chi-square value: {chi2_class_survival[0]}")
print(f"P-value: {chi2_class_survival[1]}")
```

```
Chi-square Test for Passenger Class and Survival:
Chi-square value: 6.693869422819262
P-value: 0.03519206276590605
```

For the chi-square test between passenger class and survival: The p-value is less than 0.05, it suggests that there is a significant relationship between passenger class and survival.